FPIX2 Specification

Version 1.8 (DRAFT)

January 7, 2002

**David Christian, Bradley Hall, Jim Hoff,
Abderrezak Mekkaoui, Raymond Yarema,
Sergio Zimmermann**

1. Introduction

FPIX2 consists of four logical sections: the core, the programming interface, the programmable registers and digital to analog converters, and the data output interface. A block diagram of the chip is shown in Figure 1.

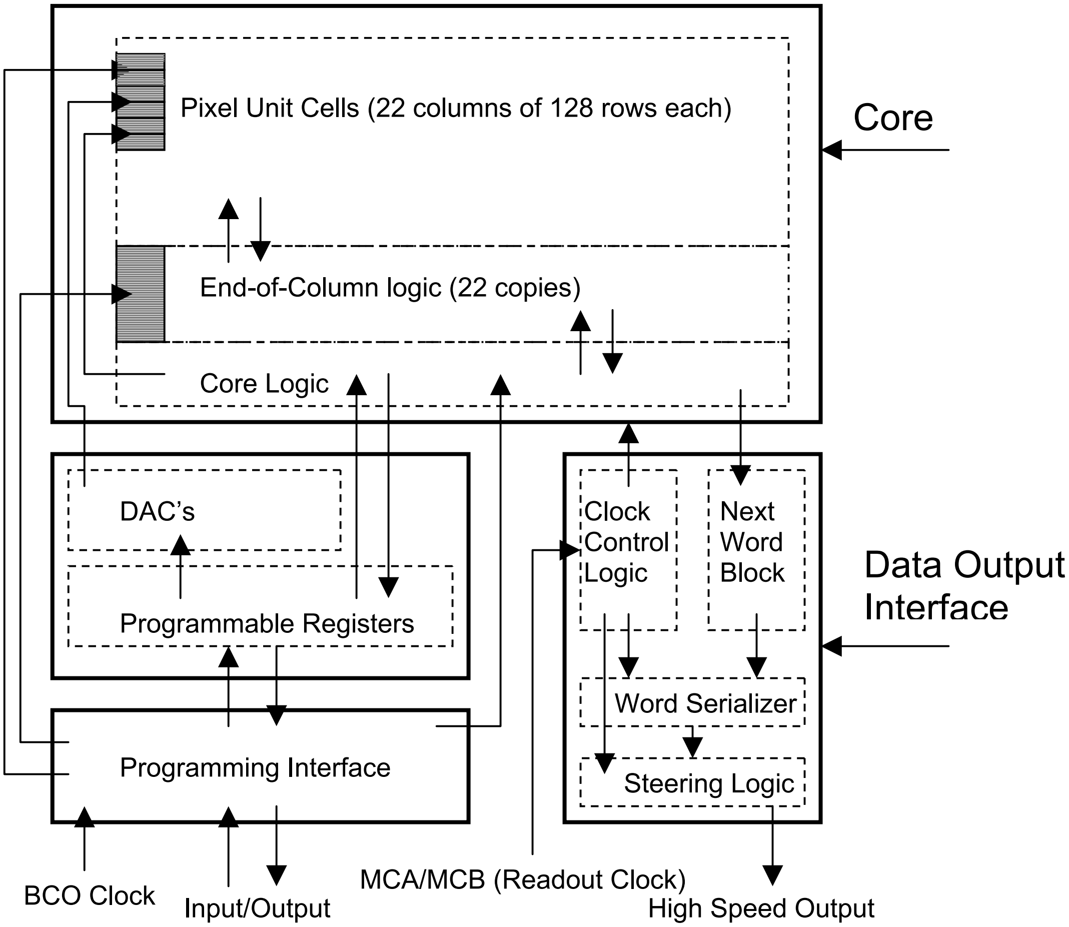

Figure 1: FPIX2 block diagram. Arrows represent control and data flow.

The core consists of the pixel unit cells, each of which contains an amplifier and a flash ADC, end-of-column logic associated with each column of pixels, and core logic, which controls the flow of data from the core to the data output interface. The programming interface accepts commands and data from a serial input bus, and, in response to commands, provides data on a serial output bus. The programmable registers are used to hold input values for the DAC's that provide currents and voltages required by the core, such as the discrimination threshold and the threshold levels for each of the FADC bits. The data output interface accepts data from the core, serializes the data, and transmits it off chip using a point-to-point protocol. All I/O (except the test signal inject) is differential and uses Low Voltage Differential Signaling (LVDS), as illustrated below. See Tables 1 and 2 for a listing of all of the FPIX2 wire bonding pads.

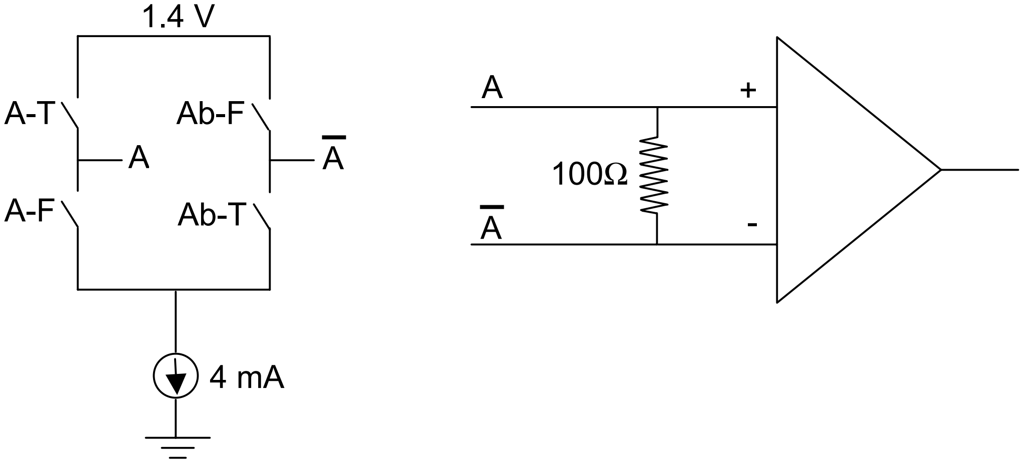

Figure 2: The drawing on the left is a simplified sketch of the FPIX2 LVDS drivers. When the signal being driven is TRUE, switches A-T and Ab-T are closed, and switches A-F and Ab-F are open. The voltage of A is 1.4V, and when the signals are terminated as shown on the right hand side of the figure, 4mA flows through the 100Ω external resistor to Ab, making its voltage 1.0V. When the signal is FALSE, switches A-F and Ab-F are closed, A-T and Ab-T are open, and the voltages and direction of current flow are reversed.

16 Analog Pads	8 Programming Interface Pads	30 Data Output Interface Pads	15 Miscellaneous Pads
5 – Bypass Capacitors	2 – Shift Control	12 – Serial Output	2 – BCO Clock
1 – Test signal Inject	2 – Shift In	2 – Data Latch (Output) Clock	1 – n Side Guard Ring Bias
3 – Power	2 – Shift Out	2 – MCA	4 – Power
3 – Ground	1 – Power	2 – MCB	4 – Ground
3 – Substrate	1 – Ground	6 – Power	2 – Fire Fighter Reset
1 – Reference Resistor		6 – Ground	2 – Operational Reset

Table 1: 69 FPIX2 wire-bonding pads by functional group. One pad has been reserved for last-minute use, so the total number of pads is 70.

Pad Number	Pad Name & Description	Pad Number	Pad Name & Description
1	V_{DDA} (+2.5V) Analog Bias	36	MCAb
2	V_{SSA} (Gnd) Analog Ground	37	MCA
3	V_{SSA2} (Gnd) Analog Ground	38	MCBb
4	InjectIn	39	MCB
5	V_{DDD} (+2.5V) Core Bias	40	V_{SSD} (Gnd) DOI Ground
6	V_{SSD} (Gnd) Core Ground	41	V_{DDD} (+2.5V) DOI Bias
7	V_{DDD} (+2.5V) PI Bias	42	Out6b
8	V_{SSD} (Gnd) PI Ground	43	Out6
9	Reference Resistor	44	Out5b
10	Bypass 1	45	Out5
11	Bypass 2	46	V_{SSD} (Gnd) DOI Ground
12	Bypass 3	47	V_{DDD} (+2.5V) DOI Bias
13	Bypass 4	48	Out4b
14	Bypass 5	49	Out4
15		50	Out3b
16	n-side sensor guard ring	51	Out3
17	V_{DDA} (+2.5V) Analog Bias	52	V_{SSD} (Gnd) DOI Ground
18	V_{SSA} (Gnd) Analog Ground	53	V_{DDD} (+2.5V) DOI Bias
19	V_{SSA2} (Gnd) Analog Ground	54	Out2b
20	Firefighter Resetb	55	Out2
21	Firefighter Reset	56	V_{SSD} (Gnd) DOI Ground
22	BCO Clockb	57	V_{DDD} (+2.5V) DOI Bias
23	BCO Clock	58	Out1b
24	V_{SSD} (Gnd) Core Ground	59	Out1
25	V_{DDD} (+2.5V) Core Bias	60	V_{SSD} (Gnd) DOI Ground
26	V_{DDD} (+2.5V) Core Bias	61	V_{DDD} (+2.5V) DOI Bias
27	V_{SSD} (Gnd) Core Ground	62	DLCLKb
28	Shift Controlb	63	DLCLK
29	Shift Control	64	V_{SSD} (Gnd) DOI Ground
30	Shift Inb	65	V_{DDD} (+2.5V) DOI Bias
31	Shift In	66	V_{SSD} (Gnd) Core Ground
32	Shift Outb	67	V_{DDD} (+2.5V) Core Bias
33	Shift Out	68	V_{SSA2} (Gnd) Analog Ground
34	Operation Resetb	69	V_{SSA} (Gnd) Analog Ground
35	Operation Reset	70	V_{DDA} (+2.5V) Analog Bias

Table 2: FPIX2 wire-bonding pads. Analog Bias powers the analog section of the pixel unit cells, and the DAC's. PI stands for "Programming Interface" and DOI stands for "Data Output Interface."

2. FPIX Core

The FPIX Core consists of an array of 22 columns of 128 rows of active pixels. Each column has associated end of column logic. The core communicates with other FPIX2 logical blocks through the core logic… (this section will be finished last). This section will include the statement that FPIX2 does not require any phase or frequency relationship between the BCO clock and the readout clocks (MCA/MCB).

3. Programming Interface

The programming interface provides a means for the user to control the operation of FPIX2, and to load and read back the contents of any of the programmable registers. Serial commands are input to the programming interface using the “shift control” and “shift in” lines. When “shift control” is high, “shift in” is latched into the input register of the programming interface on the falling edge of the BCO clock. After a “read” command, the contents of the requested register are output on “shift out.” “Shift control” must also be kept high after a “read” command while data is being output on “shift out.” As indicated in Figure 3, there is a one-cycle delay before the output appears. Data is shifted out on the rising edge of the BCO clock.

The programming interface will respond to all broadcast (wild chip address = 10101) commands, and to all commands in which the chip address matches the contents of the “chip address register” (set by internal wire bonds). Each command consists of 5 bits of chip address, followed by a 5-bit register number, and a 3-bit instruction code. For Write commands only, the instruction code is followed by data, which is written to the specified register. With one exception, all “set,” “reset,” and “default” commands affect a register for an entire BCO clock period, starting on the rising edge immediately after the last instruction bit is latched. The exception is the “AqBCO, set” (Acquire current BCO number) command, which is executed on the first negative going BCO clock edge after “shift control” goes low. The FPIX2 command format is illustrated in Figure 3, and the instruction codes are listed in Table 3 below.

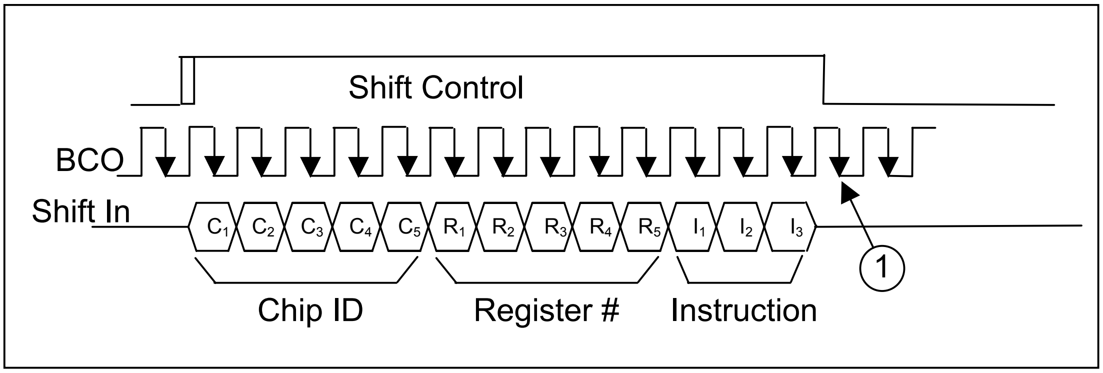

Figure 3: Programming Interface Input Format. The "Aquire BCO" set command is executed at the clock edge indicated by the number 1 in a circle.

Instruction	Code
Write (followed by 2, 8, or 2816 bits of data)	001
Set (all bits in register = 1)	010
Read	100
Reset (all bits in register = 0)	101
Default (set register to default value)	110

Table 3: Programming Interface Instructions.

4. Programmable Registers and DAC's

The registers are listed in Table 4. The table also lists how the Firefighter Reset, Operation Reset, and the Smart Programming Reset affect each register.

The first 14 registers in Table 4 (Ibp – Vth7; note that Vth1 is not used) are all 8-bit registers. These registers hold the values that are input to the digital to analog converters. All of the threshold DAC's cover the same range, and in each case the least significant bit corresponds to approximately 200 electrons at the input of a pixel amplifier.

AqBCO is also an 8-bit register. When AqBCO is Set, the value of the BCO counter is loaded into the AqBCO register. This register can be read at any later time to verify BCO synchronization. BCO synchronization can also be checked without reading the AqBCO register if the AqBCO Set command is timed so that the BCO counter value latched should equal zero. If the AqBCO register contents are not zero, a bit is set in the sync/status word (see section 5).

The “active lines” register, Alines, is a 2-bit register. This register, which is implemented using redundant logic designed to be immune to single event upset, determines the data output configuration (see section 5).

Kill and Inject are serpentine registers running up and down the pixel columns. Each register has one bit in each pixel unit cell (22 x 128 = 2816 bits in each register). Kill=1 opens a switch at the output of the pixel discriminator, effectively killing the pixel. Inject=1 closes a switch connecting the Test Signal Inject line to the charge injection capacitor associated with a pixel.

SendData and RejectHits are both SEU tolerant single bit registers. Setting RejectHits=1 inhibits the core from accepting any more pixel hits (data already latched is not affected). Setting SendData=0 disables the core readout. If data is being read out when SendData is changed from 1 to 0, up to two data words may be lost (either one or none on the transition from 1 to 0, and either one or none on the transition back from 0 to 1).

Three of the register addresses do not correspond to physical registers. WildReg (10101) is used in broadcast commands that, for instance, set all 8-bit registers to their default values. Two “registers” are actually commands, which are executed when the “register” is Set. SCR is the Smart Core Reset. This command clears all pixels and end-of-column logic, and resets the BCO counter to zero. SPR is the Smart Programming Reset. SPR resets the bias currents to default values and all thresholds to default (safe) values, but does not affect the data output configuration, the Kill and Inject registers, or SendData or RejectHits. SCR and SPR are discussed more fully in Section 6, which also contains a description of the two hardware resets (“Firefighter Reset” and “Operation Reset”).

All of the registers except Kill and Inject are loaded least significant bit (b0) first. All of the registers except Kill and Inject can be read non-destructively at any time. After a read command, the requested register contents are copied to a shadow register, then shifted out, most significant bit first. Note this bit order is opposite to the order used to load the register.

Kill and Inject are loaded by shifting data into the register in the order illustrated in Figure 4. The bit intended for column 21, row 0, is input first. The bit intended for column 0, row 0, is input last. Kill and Inject cannot be read non-destructively, as no shadow register is implemented. After a read command, data from Kill or Inject appears on “shift out” in the same order that it was loaded – (21,0) first. As the data is read out, it is also shifted back through the entire shift register, so that at the end of the read operation (if “shift control” is lowered just after the last bit is shifted out), the register contents are restored.

Register Name	Address	Firefighter Reset	Operation Reset	Smart Prog Reset	Notes
	00000_2 (0)				Forbidden
Ibp	00001_2 (1)	Default=80	Unchanged	Default	
Ibp1	00010_2 (2)	Default=80	Unchanged	Default	
Iff	00011_2 (3)	Default=70	Unchanged	Default	
Ifb	00100_2 (4)	Default=0	Unchanged	Default	
Vref	00101_2 (5)	Default=130	Unchanged	Default	
Vfb	00110_2 (6)	Default=0	Unchanged	Default	
Vth0	00111_2 (7)	Default=0	Unchanged	Default	
Vth1	01000_2 (8)				Unused in FPIX2
Vth2	01001_2 (9)	Default=0	Unchanged	Default	
Vth3	01010_2 (10)	Default=0	Unchanged	Default	
Vth4	01011_2 (11)	Default=0	Unchanged	Default	
Vth5	01100_2 (12)	Default=0	Unchanged	Default	
Vth6	01101_2 (13)	Default=0	Unchanged	Default	
Vth7	01110_2 (14)	Default=0	Unchanged	Default	
AqBCO	01111_2 (15)	Default=0	Unchanged	Default	
Alines	10000_2 (16)	00_2 (0)	Unchanged	Unchanged	**Ignores WildReg**
Kill	10001_2 (17)	“no kill”	Unchanged	Unchanged	**Ignores WildReg**
Inject	10010_2 (18)	“no inj”	Unchanged	Unchanged	**Ignores WildReg**
SendData	10011_2 (19)	0(noSend)	Unchanged	Unchanged	**Ignores WildReg**
RejectHits	10100_2 (20)	1(reject)	Unchanged	Unchanged	**Ignores WildReg**
WildReg	10101_2 (21)				
	10110_2 (22)				Unused in FPIX2
	10111_2 (23)				Unused in FPIX2
SPR	11000_2 (24)	Unaffected	Unaffected	Unaffected	**Ignores WildReg**
	11001_2 (25)				Unused in FPIX2
	11010_2 (26)				Unused in FPIX2
	11011_2 (27)				Unused in FPIX2
SCR	11100_2 (28)	Unaffected	Unaffected	Unaffected	**Ignores WildReg**
	11101_2 (29)				Unused in FPIX2
	11110_2 (30)				Unused in FPIX2
	11111_2 (31)				Forbidden

Table 4: FPIX2 Programmable Registers.

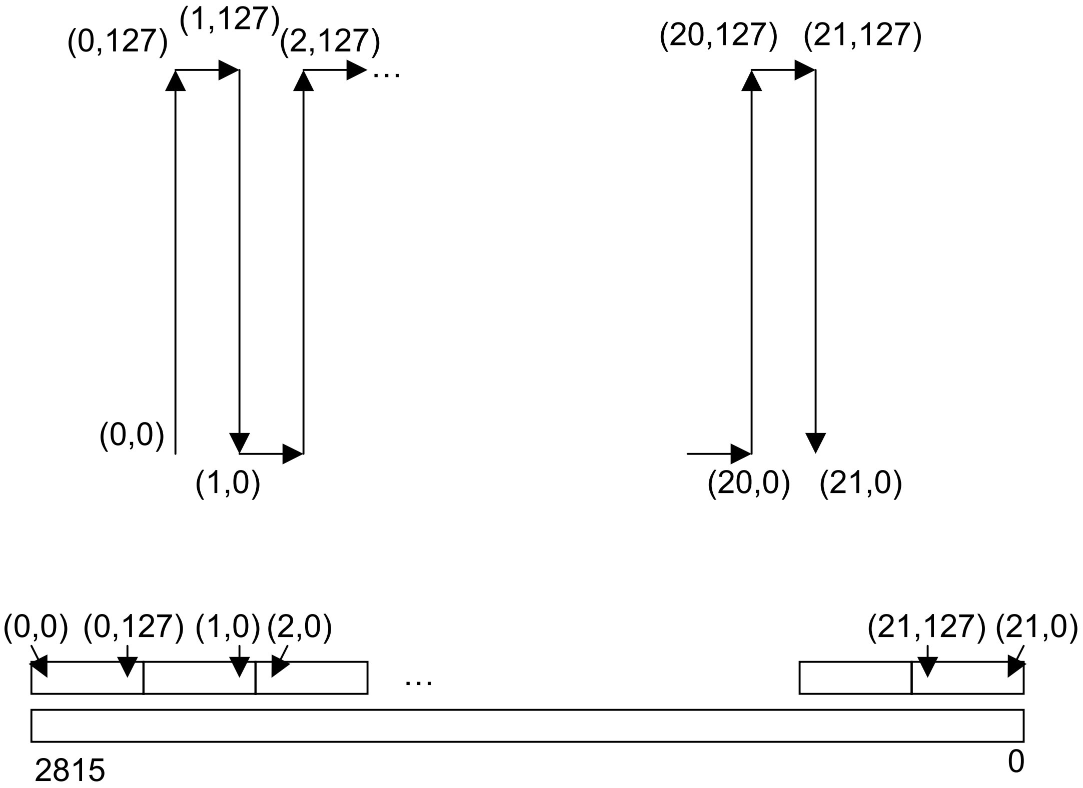

Figure 4: The Kill and Inject registers are loaded in the order indicated above. The bit intended for pixel column 21, row 0, is shifted in first. The bit intended for (0,0) is shifted in last. The registers are read back in the same order: (21,0) appears on "shift out" first, (0,0) appears last.

5. Data Output Interface

There are four functional blocks in the data output interface: the clock control logic, the next word block, the word serializer, and the steering logic. The relationship between these blocks is shown in Figure 5.

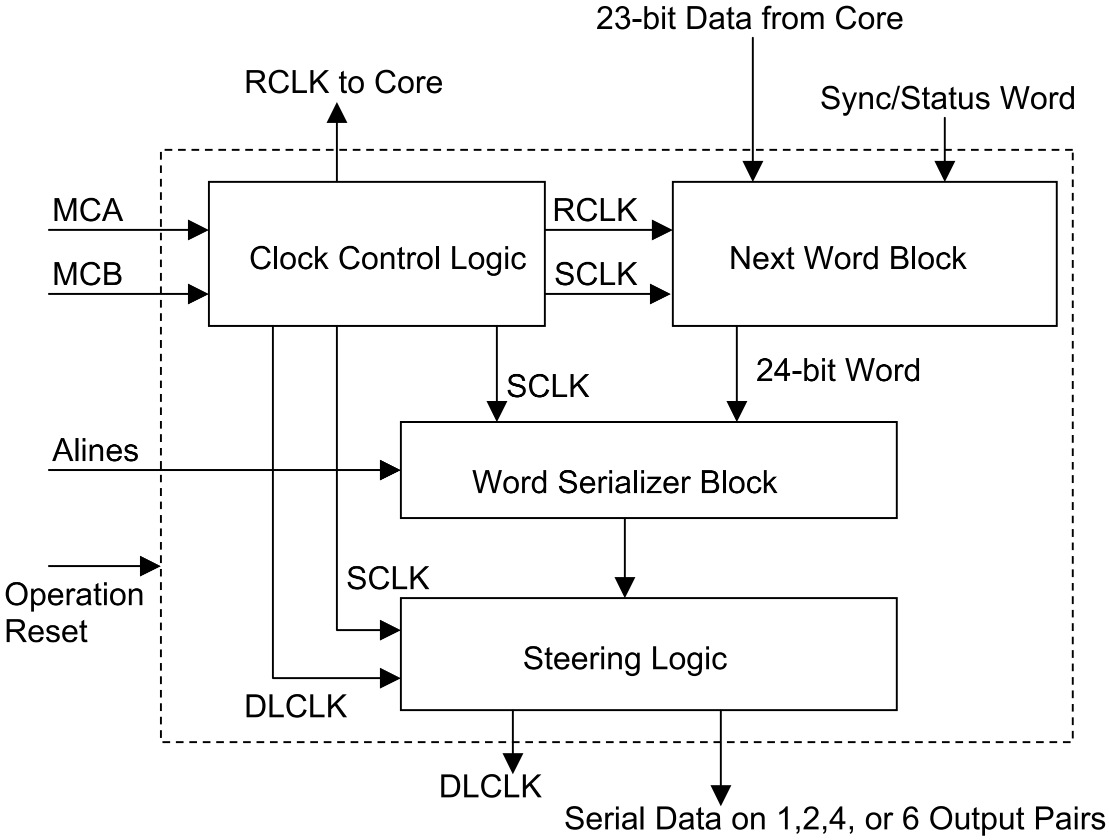

Figure 5: Data Output Interface block diagram.

The clock control logic block receives two clock signals from the Pixel Data Combiner Board (PDCB). These two clocks (Master Clock A and Master Clock B) are bussed to all FPIX chips in a module. MCA and MCB are both nominally 68.8 MHz clocks, with MCB lagging MCA by 90^{o}. The clock control logic block uses MCA and MCB to derive the two clocks used internally to control readout (RCLK and SCLK), as well as the output Data Latch Clock (DLCLK), which is output along with the serialized data. The PDCB may adjust the phase relation of MCA and MCB as necessary to maximize the length of time that data received from FPIX2 is valid.

The Serial Clock (SCLK) is derived from MCA and MCB using a simple Exclusive OR. The core Readout Clock (RCLK) used by the FPIX core is derived using a counter. SCLK is twice the frequency of MCA and MCB. The frequency of RCLK depends on the number of active data output lines (see table below), and is given by dividing SCLK by the number of bits serialized on each output path. The relationship between SCLK, the number of active lines, and RCLK insures that data is output from the chip at the same rate as it is read out of the core. This means that no extra buffer memory is required in the data output interface.

The output Data Latch Clock (DLCLK), which is equal in frequency to MCA, is derived from SCLK. Output data is asserted on the positive going edges of SCLK; DLCLK transitions occur on the negative going edges of SCLK.

Configuration	SCLK Frequency	RCLK Frequency
6 output pairs	(MCA) × 2	(MCA) ÷ 2
4 output pairs	(MCA) × 2	(MCA) ÷ 3
2 output pairs	(MCA) × 2	(MCA) ÷ 6
1 output pair	(MCA) × 2	(MCA) ÷ 12

Table 5: Dependence of SCLK and RCLK on data output configuration.

The next word block selects the source of the next word to be input to the word serializers. If “core talking” is asserted and it has been asserted for at least one RCLK cycle, then the “core data word” is selected and latched into flip flops in the output stage of the next word block. Otherwise, the “sync word” is selected and latched into the same flip-flops. This data transfer takes place on the falling edge of RCLK. When the core horizontal token reaches column number 21, there is at least one RCLK cycle in which “core talking” is not asserted. When the horizontal token is launched again (in response to “core has data”), one more RCLK cycle is required before valid data is delivered from the core to the data output interface. This guarantees that the “sync word” will be output at least twice every time that the FPIX2 readout scans through all 22 columns. As illustrated below, the next word block adds a word mark bit, which is guaranteed to be one, to the data. The PDCB uses the sync word to establish the location of the 24-bit word boundary. It can distinguish between the sync word and core data because the sync word has 13 zeros in bits 1-13. The word mark bit, and the pixel column number encoding, ensures that core data can never have 13 consecutive bits equal to zero, either within a word, or across two data words. The most significant 10 bits (b14 – b23) of the sync word are reserved for status and error codes. So far, five of these bits have been assigned (see Table 6).

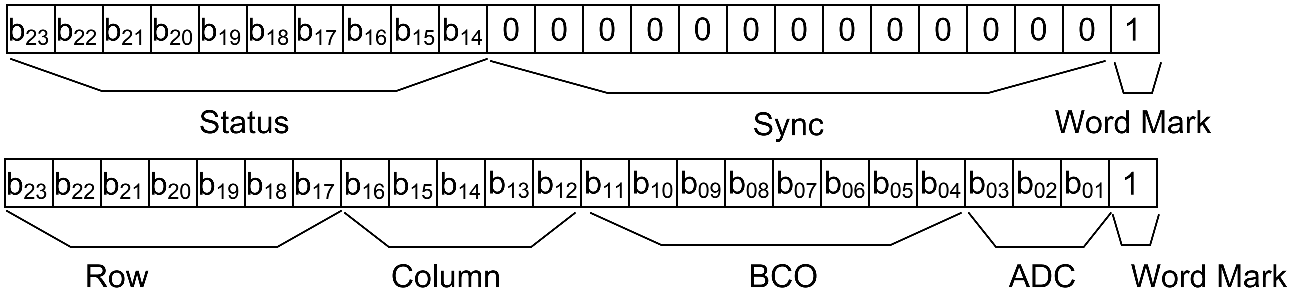
b_{23} b_{22} b_{21} b_{20} b_{19} b_{18} b_{17} b_{16} b_{15} b_{14} 0 0 0 0 0 0 0 0 0 0 0 0 0 1
Status
Sync
Word Mark
b_{23} b_{22} b_{21} b_{20} b_{19} b_{18} b_{17} b_{16} b_{15} b_{14} b_{13} b_{12} b_{11} b_{10} b_{09} b_{08} b_{07} b_{06} b_{05} b_{04} b_{03} b_{02} b_{01} 1
Row
Column
BCO
ADC
Word Mark

Figure 6: Format of sync word (top) and data word (bottom).

Bit Number	Meaning	Bit Number	Meaning
23	SendData	18	Not yet assigned
22	RejectHits	17	Not yet assigned
21	Alines-b1	16	Not yet assigned
20	Alines-b0	15	Not yet assigned
19	AqBCO ≠ 0	14	Not yet assigned

Table 6: Status bit assignment in sync word.

Col. #	Code	Col. #	Code	Col. #	Code	Col. #	Code
0	11011	6	00110	12	11110	18	10010
1	00001	7	00111	13	01101	19	10011
2	00010	8	11010	14	01110	20	11101
3	00011	9	01001	15	01111	21	10101
4	10111	10	01010	16	11001		
5	00101	11	01011	17	10001		

Table 7: Column numbering code.

The word serializer block serializes the data for output. On the falling edge of RCLK, the word held in the flip flops at the output of the next word block is latched into the serializer flip flops. This data is output serially to the steering logic in 1, 2, 4, or 6 parallel paths depending on the status of the "active lines" register (Alines), which controls the data output configuration.

Alines Contents	Number of Output Pairs
00	1
01	2
10	4
11	6

Table 8: Active lines register code.

The steering logic drives the output data clock and the serial output data off chip. DLCLK is one half the frequency of SCLK and is phased so that its edges fall one half way between the edges of the data lines.

6. Resets

The FPIX2 has two hardware resets (“fire fighter reset” and “operation reset”) and two software resets (“smart core reset” and “smart programming reset”). It is expected that all chips on a sensor module will share the hardware reset lines, so that all chips will be reset at once. The software reset commands may be broadcast to all chips on a module, or sent to one chip only.

When a “smart core reset” is received, the BCO counter is reset to zero, all end-of-column logic is cleared, and all pixel hits are cleared.

When a “smart programming reset” is received, all of the registers that hold values input to DAC’s are reset to their default values, and the AqBCO register is cleared. Since the default value for all of the threshold registers is zero (maximum threshold), this effectively disables all pixels even though RejectHits is unchanged.

When an “operation reset” is received, the data output interface is reset. This consists of zeroing all of the word serializer registers, halting RCLK and SCLK, then starting RCLK and SCLK again such that when data is shifted out of the chip, the word mark bit is accompanied by a 0 → 1 transition of DLCLK. This is necessary because the FPGA in the PDCB uses two sets of input latches, one which latches on negative going DLCLK transitions, and one which latches on positive going transitions. The FPGA micro code requires that the word mark bit be latched by a 0 → 1 transition of DLCLK.

When a “fire fighter reset” is received, the FPIX2 is reset to a “safe” mode. The effect is the same as a simultaneous “smart core reset”, “smart programming reset”, and “operation reset.” In addition, the “fire fighter reset” resets Alines, Kill, Inject, SendData, and RejectHits. This means that the number of active output pairs is reset to one, no pixel unit cell is killed and none connected to the charge injection signal, data output from the core is inhibited, and all pixel unit cells are set to ignore new hits.

The various reset actions are summarized in Table 4.

7. Procedures

It is anticipated that the following procedure will be followed when turning on an FPIX2:

1) Fire Fighter Reset – place FPIX2 in a “safe” mode.
2) Write,Alines,# – configure the Data Output Interface (DOI) to use the proper number of output lines (if more than one pair is used).

3) Operation Reset – reset the DOI (required only if Alines was just changed).
4) Wait for sync word to be received, confirming DOI configuration.
5) Write,Vth0 – Vth7, # – download the thresholds.
6) Write,…,# – download any other register values required (pixels to be killed, bias currents, Vref,…).
7) Write,RejectHits,0 – enable the core.
8) Set,SCR – smart core reset (see below for timing necessary to insure system wide BCO number synchronization).
9) Set,SendData – enable the data readout.

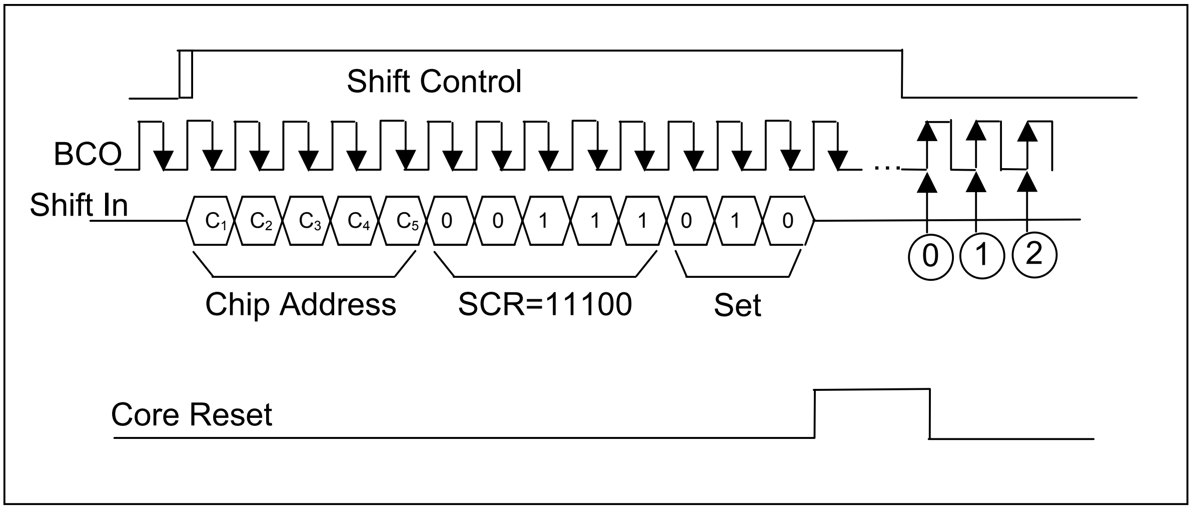

Figure 7: The "Core Reset" line is asserted on the rising edge of the BCO clock following the last bit of the SCR,Set command. It stays active until just after the first BCO clock rising edge after "shift control" has been lowered. The next rising edge of the BCO clock will increment the BCO counter from zero to one, so the BCO numbers will be assigned as indicated by the numbers in circles.

If an FPIX2 loses data output synchronization, then an operation reset will recover synchronization without requiring the core or programmable registers to be reset.

If an FPIX2 loses BCO synchronization, then a smart core reset will recover synchronization without requiring the DOI or programmable registers to be reset. As shown in Figure 7, core data will be lost until shift control is dropped at the appropriate time to restart the BCO counter at zero in sync with the rest of the system. The core will become active approximately 5 ns into BCO period 0.